UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/819,772	03/28/2001	Michael Petrov	02509/90	2624

26646 7590 01/29/2008

KENYON & KENYON LLP
ONE BROADWAY
NEW YORK, NY 10004

EXAMINER	
CUNNINGHAM, GREGORY F	
ART UNIT	**PAPER NUMBER**
2624	

MAIL DATE	DELIVERY MODE
01/29/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/819,772
Filing Date: March 28, 2001
Appellant(s): PETROV ET AL.

Kenyon & Kenyon LLP
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed December 19, 2007 appealing from the Office action mailed August 3, 2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

No evidence is relied upon by the examiner in the rejection of the claims under appeal.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

I. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

II. Claims 2-10, 55-63 and 114-117 are rejected under 35 U.S.C. 102(b) as being anticipated by MatLab Primer, hereinafter MatLab.

A. MatLab anticipates claim 5, "A method for restoring a previous version of a three dimensional mesh model on a computer system comprising:

retrieving a stored copy of an earlier state [page 3 at 'Saving a session.

When one logs out or exits MATLAB all variables are lost. However, invoking the command save before exiting causes all variables to be written to a non-human-readable diskfile named matlab.mat. When one later reenters MATLAB, the command load will restore the workspace to its former state.'] of the three dimensional mesh model [page 15 at '18. Graphics.

MATLAB can produce planar plots of curves, 3-D plots of curves, 3-D mesh surface plots, and 3-D faceted surface plots. The primary commands for these facilities are plot, plot3, mesh, and surf, respectively. An introduction to each of these is given below. To preview some of these capabilities, enter the command demo and select some of the graphics options.']; and [page 18 at '3-D mesh and surface plots.

Three dimensional wire mesh surface plots are drawn with the command mesh. The command mesh(z) creates a three-dimensional perspective plot of the elements of the matrix z. The mesh surface is defined by the z-coordinates of points above a rectangular grid in the x-y plane. Try mesh(eye(10)).

Similarly, three dimensional faceted surface plots are drawn with the command surf. Try surf(eye(10)).

To draw the graph of a function z = f(x; y) over a rectangle, one first defines vectors xx and yy which give partitions of the sides of the rectangle. With the function meshgrid one then creates a matrix x, each row of which equals xx and whose column length is the length of yy, and similarly a matrix y, each column of which equals yy, as follows:

[x,y] = meshgrid(xx,yy);

One then computes a matrix z, obtained by evaluating f entrywise over the matrices x and y, to which mesh or surf can be applied. You can, for example, draw the graph of

$z = e^{-x2-y2}$ over the square [-2; 2] x [-2; 2] as follows (try it):

xx = -2:.2:2;

yy = xx;[x,y] = meshgrid(xx,yy);

z = exp(-x.^2 - y.^2);

mesh(z)

One could, of course, replace the first three lines of the preceding with

[x,y] = meshgrid(-2:.2:2, -2:.2:2);'] on the computer system [page ii, second para. at 'computer'];

retrieving an ordered list of operations on the computer system [page 9, at '12. M-files.

MATLAB can execute a sequence of statements stored in diskfiles. Such files are called \M-files" because they must have the file type of \.m" as the last part of their filename. Much of your work with MATLAB will be in creating and refining M-files. M-files are usually created using your local editor.

There are two types of M-files: script files and function files. Script files.

A script file consists of a sequence of normal MATLAB statements. If the file has the filename, say, rotate.m, then the MATLAB command rotate will cause the statements in the file to be executed. Variables in a script file are global and will change the value of variables of the same name in the environment of the current MATLAB session.

Script files may be used to enter data into a large matrix; in such a file, entry errors can be easily corrected. If, for example, one enters in a diskfile data.m

```
A = [
1 2 3 4
5 6 7 8
];
```

then the MATLAB statement data will cause the assignment given in data.m to be carried out. However, it is usually easier to use the MATLAB function load (see section 2).

An M-file can reference other M-files, including referencing itself recursively.']; and

performing at least some of the operations in the ordered list of operations on the retrieved copy of the three dimensional mesh model [see page 18, supra,

'Three dimensional wire mesh surface plots are drawn with the command mesh. The command mesh(z) creates a three-dimensional perspective plot of the elements of the matrix z.

The mesh surface is defined by the z-coordinates of points above a rectangular grid in the x-y plane. Try mesh(eye(10)).

Similarly, three dimensional faceted surface plots are drawn with the command surf. Try surf(eye(10)).

To draw the graph of a function z = f(x; y) over a rectangle, one first defines vectors xx and yy which give partitions of the sides of the rectangle. With the function meshgrid one then creates a matrix x, each row of which equals xx and whose column length is the length of yy, and similarly a matrix y, each column of which equals yy, as follows:

[x,y] = meshgrid(xx,yy);

One then computes a matrix z, obtained by evaluating f entrywise over the matrices x and y, to which mesh or surf can be applied. You can, for example, draw the graph of

$z = e^{-x2-y2}$ over the square [-2; 2] x [-2; 2] as follows (try it):

xx = -2:.2:2;

yy = xx;[x,y] = meshgrid(xx,yy);

z = exp(-x.^2 - y.^2);

mesh(z)']

One could, of course, replace the first three lines of the preceding with

[x, y] = meshgrid(-2:.2:2, -2:.2:2);

wherein the ordered list of operations contains the operations [M-file] which if performed in order [an executed M-file] on the earlier state of the three dimensional mesh model [page 18, 3D mesh] would result in a current state of the three dimensional mesh model [see page 9, M-files and page 18, 3D mesh as given supra, whereby]" [as detailed].

Thus Matlab expressly and/or inherently anticipates claim 5 as detailed supra, particularly since the user is encouraged to work at the computer while reading the Matlab Primer and freely experiment with the examples and to draw a 3-D mesh graph.

B. MatLab anticipates claim 6, "The method of claim 5 wherein each operation is performed in the same order in which it was originally placed in the ordered list [ordered list of M-file and given 3D mesh example, see pages 9 and 18, supra for claim 5]" supra for claim 5 and [as detailed].

C. MatLab anticipates claim 7, "The method of claim 6 further comprising the step of." rendering the retrieved copy of the three dimensional mesh model to a display device after each operation is performed [page 3 at '4. Statements, expressions, and variables; saving a session. MATLAB is an expression language; the expressions you type are interpreted and evaluated. MATLAB statements are usually of the form

variable = expression, or simply

expression

Expressions are usually composed from operators, functions, and variable names. Evaluation of the expression produces a matrix, which is then displayed on the screen and assigned to the variable for future use. If the variable name and = sign are omitted, a variable ans (for answer) is automatically created to which the result is assigned.

A statement is normally terminated with the carriage return. However, a statement can be continued to the next line with three or more periods followed by a carriage return. On the other hand, several statements can be placed on a single line if separated by commas or semicolons.

If the last character of a statement is a semicolon, the printing is suppressed, but the assignment is carried out. This is essential in suppressing unwanted printing of intermediate results.

MATLAB is case-sensitive in the names of commands, functions, and variables. For example, solveUT is not the same as solveut.

The command who (or whos) will list the variables currently in the workspace. A variable can be cleared from the workspace with the command clear variablename. The command clear alone will clear all nonpermanent variables.

The permanent variable eps (epsilon) gives the machine unit roundo_|about 10□□16 on most machines. It is useful in specifying tolerences for convergence of iterative processes.

A runaway display or computation can be stopped on most machines without leaving MATLAB with CTRL-C (CTRL-BREAK on a PC). Saving a session.

When one logs out or exits MATLAB all variables are lost. However, invoking the command save before exiting causes all variables to be written to a non-human-readable diskfile named matlab.mat. When one later reenters MATLAB, the command load will restore the workspace to its former state.'] and [page 15 at '18. Graphics.

MATLAB can produce planar plots of curves, 3-D plots of curves, 3-D mesh surface plots, and 3-D faceted surface plots. The primary commands for these facilities are plot, plot3, mesh, and surf, respectively. An introduction to each of these is given below. To preview some of these capabilities, enter the command demo and select some of the graphics options.' Wherein leaving the semicolon off renders each to the printer.] and [page 18 at 3-D mesh and surface

plots, given supra for claim 5, wherein the plot command renders to a display device]" supra for claim 6 and [as detailed].

D. MatLab anticipates claim 8, "The method of claim 6 wherein the ordered list of operations is filtered to exclude at least one record [page 18 at 'Completely analogous to plot in two dimensions, the command plot3 produces curves in three dimensional space. If x, y, and z are three vectors of the same size, then the command plot3(x,y,z) will produce a perspective plot of the piecewise linear curve in 3-space passing through the points whose coordinates are the respective elements of x, y, and z. These vectors are usually defined parametrically. For example, t=.01:.01:20*pi; x=cos(t); y=sin(t); z=t.^3; plot3(x,y,z)
will produce a helix which is compressed near the x-y plane (a \slinky"). Try it.

Just as for planar plots, a title and axis labels (including zlabel) can be added. The features of axis command described there also hold for 3-D plots; setting the axis scaling to prescribed limits will, of course, now require a 6-vector.' Wherein limiting the axis scale will limit (filter) the plotted values for (at least one record) if not more depending on the set limit values]" supra for claim 6 and [as detailed].

E. MatLab anticipates claim 9, "The method of claim 8 wherein the at least one excluded record is at an end of the list [Just as for planar plots, a title and axis labels (including zlabel) can be added. The features of axis command described there also hold for 3-D plots; setting the axis scaling to prescribed limits will, of course, now require a 6-vector.' Wherein limiting the axis scale will limit (filter) the plotted values for (at least one record at the end of a list) if not more depending on the set limit values]" supra for claim 8 and [as detailed].

(Examiner's note: Beyond this MatLab Primer, MatLab also has many toolboxes with filtering functions, and commands also to set floor, ceiling, domain and range parameters.)

F. MatLab anticipates claim 10, "The method of claim 8 wherein the at least one excluded record is at least one record removed from an end of the list [Just as for planar plots, a title and axis labels (including zlabel) can be added. The features of axis command described there also hold for 3-D plots; setting the axis scaling to prescribed limits will, of course, now require a 6-vector.' Wherein limiting the axis scale will limit (filter) the plotted values for (at least one record removed from end of list) if not more depending on the set limit values]" supra for claim 8 and [as detailed].

(Examiner's note: Beyond this MatLab Primer, MatLab also has many toolboxes with filtering functions, and commands also to set floor, ceiling, domain and range parameters.)

G. Per independent claim 58, this is directed to an article of manufacture for performing the method of independent claim 5, and therefore is rejected to independent claim 5.

H. Per dependent claims 59-63, these are directed to an article of manufacture for performing the method of dependent claims 6-10, and therefore are rejected to dependent claims 6-10.

J. MatLab anticipates claim 115, "A method for managing a three dimensional mesh model on a computer system, comprising:

storing a copy of a first state of the three dimensional mesh model on the computer system [page 3 at 'Saving a session.

When one logs out or exits MATLAB all variables are lost. However, invoking the command save before exiting causes all variables to be written to a non-human-readable diskfile

named matlab.mat. When one later reenters MATLAB, the command load will restore the workspace to its former state.'] of the three dimensional mesh model [page 15 at '18. Graphics.

MATLAB can produce planar plots of curves, 3-D plots of curves, 3-D mesh surface plots, and 3-D faceted surface plots. The primary commands for these facilities are plot, plot3, mesh, and surf, respectively. An introduction to each of these is given below. To preview some of these capabilities, enter the command demo and select some of the graphics options.']; and [page 18 at '3-D mesh and surface plots.

Three dimensional wire mesh surface plots are drawn with the command mesh. The command mesh(z) creates a three-dimensional perspective plot of the elements of the matrix z. The mesh surface is defined by the z-coordinates of points above a rectangular grid in the x-y plane. Try mesh(eye(10)).

Similarly, three dimensional faceted surface plots are drawn with the command surf. Try surf(eye(10)).

To draw the graph of a function z = f(x; y) over a rectangle, one first defines vectors xx and yy which give partitions of the sides of the rectangle. With the function meshgrid one then creates a matrix x, each row of which equals xx and whose column length is the length of yy, and similarly a matrix y, each column of which equals yy, as follows:

[x,y] = meshgrid(xx,yy);

One then computes a matrix z, obtained by evaluating f entrywise over the matrices x and y, to which mesh or surf can be applied. You can, for example, draw the graph of

$z = e^{-x2-y2}$ over the square [-2; 2] x [-2; 2] as follows (try it):

xx = -2:.2:2;

yy = xx;[x,y] = meshgrid(xx,yy);

z = exp(-x.^2 - y.^2);

mesh(z)

One could, of course, replace the first three lines of the preceding with [x,y] = meshgrid(-2:.2:2, -2:.2:2);'];

performing operations on the three dimensional mesh model, wherein the three dimensional mesh model is in a second state after performing the operations [MatLab's M-file and 3-D mesh, see pages 9 and 18];

storing a record of each of the operations in an ordered list on the computer system [M-file, page 9]; and

reapplying at least some of the operations stored in the ordered list to the stored first state of the three dimensional mesh model [The editing of M-files to modify the ordered list of the M-file, see page 13 at 'Managing M-files'; and

page 9, bottom at "An M-file can reference other M-files, including referencing itself recursively.'], wherein the three dimensional mesh model is in a third state after reapplying the at least some of the operations [Editing M-files via page 13 'Managing M-files' and exemplified on page 18 at 'One could, of course, replace the first three lines of the preceding with [x,y] = meshgrid(-2:.2:2, -2:.2:2); Try this plot with surf instead of mesh', whereby "a third state" is expressly and/or inherently anticipated by editing the parameter values and/or functions of the M-file]" [as detailed].

Thus Matlab expressly and/or inherently anticipates claim 115 as detailed supra, particularly since the user is encouraged to work at the computer while reading the Matlab Primer and freely experiment with the examples and to draw a 3-D mesh graph.

K. MatLab anticipates claim 2, "The method of claim 115 wherein the step of storing a record of each of the operations includes:

storing all of the parameters necessary to repeat the operations [corresponds to MatLab's save command: 'invoking the command save before exiting causes all variables to be written to a non-human-readable diskfile named matlab.mat' – page 3, and/or M-files, see page 9 given supra]" supra for claim 115 and [as detailed].

L. MatLab anticipates claim 3, "The method of claim 2 wherein the ordered list contains a record for each operation that has been previously performed on the three dimensional mesh model in the order in which it was performed [M-files and 3-D mesh, see pages 9 and 18, given supra]" supra for claim 2 and [as detailed].

M. MatLab anticipates claim 4, "The method of claim 115 wherein the step of reconstructing the three dimensional model includes:

retrieving the stored copy of the first state of the three dimensional mesh model [MatLab's load command, see 'the MATLAB command load.ext will read this file to the variable data in your MATLAB workspace. This may also be done with a script file (see section 12)' - pages 2 , 'the command load will restore the workspace to its former state' – page 3, and M-files – page 9];

retrieving the ordered list of operations [M-file – page 9, given supra]; and

performing at least one operation in the ordered list of operations on the retrieved copy of the first state of the three dimensional mesh model [M-file and 3-D mesh – pages 9 and 18, given supra]" supra for claim 115 and [as detailed].

N. Per independent claims 116 and 117, these are directed to an article of manufacture and a system, respectively, for performing the method of independent claim 115, and therefore are rejected to independent claim 115.

P. Per dependent claims 55-57 and 114, these are directed to an article of manufacture ans a system, respectively, for performing the method of dependent claims 2-4, and therefore are rejected to dependent claims 2-4.

(10) Response to Argument

I. The Appellants' opening argument "In order for a claim to be anticipated under 35 U.S.C. § 102, a single prior art reference must disclose each and every element of the claim in exactly the same way". This should not be interpreted as the single prior art reference must disclose each and every element of the claim using exactly the same claim language. More to the point by way of the first follow-on reference cited by the Appellants, " See, e.g., Lindeman Machinenfabrik v. Am. Hoist and Derrick, 730 F.2d 1452, 1458 (Fed. Cir. 1984)" which states:

Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim. *Connell v. Sears, Roebuck & Co.*, 722 F.2d 1542, 220 USPQ 193 (Fed.Cir. 1983); *SSIH Equip. S.A. v. USITC*, 718 F.2d 365, 218 USPQ 678 (Fed.Cir. 1983).

Although the reference 'Matlab' does not disclose each and every element of the claimed invention in the exact sequential order line for line verbatim as the claim, 'Matlab' does meet the conditions of anticipation by disclosure of each and every element of the claimed invention, as arranged in the claim. Matlab's functionality in the manner it carries out its operations as a user interactive, matrix-based system for scientific and engineering numeric computation and visualization ('Matlab' – page ii, 1st paragraph), by making use of how it stores, saves and reloads sessions, variables, M-files, restores a workspace to its former state, 3-D mesh surface plots, and the execution of M-files and demo files as detailed supra for the rejection of claims 5, 58 and 115 - 117.

In addition, the second follow-on reference cited by the Appellants, "MPEP § 2131", states:

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). >"When a claim covers several structures or compositions, either generically or as alternatives, the claim is deemed anticipated if any of the structures or compositions within the scope of the claim is known in the prior art." Brown v. 3M, 265 F.3d 1349, 1351, 60 USPQ2d 1375, 1376 (Fed. Cir. 2001) (claim to a system for setting a computer clock to an offset time to address the Year 2000 (Y2K) problem, applicable to records with year date data in "at least one of two-digit, three-digit, or four-digit" representations, was held anticipated by a system that offsets year dates in only two-digit formats). See also MPEP § 2131.02.< "The identical invention must be shown in as complete detail as is contained

in the ... claim." Richardson v. Suzuki Motor Co., 868 F.2d 1226, 1236, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). The elements must be arranged as required by the claim, but this is not an ipsissimis verbis test, i.e., identity of terminology is not required. In re Bond, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990). Note that, in some circumstances, it is permissible to use multiple references in a 35 U.S.C. 102 rejection.

See MPEP § 2131.01.

2131.01 Multiple Reference 35 U.S.C. 102 Rejections

Normally, only one reference should be used in making a rejection under 35 U.S.C. 102. However, a 35 U.S.C. 102 rejection over multiple references has been held to be proper when the extra references are cited to:

(A) Prove the primary reference contains an "enabled disclosure; "

(B) Explain the meaning of a term used in the primary reference; or

(C) Show that a characteristic not disclosed in the reference is inherent."

To emphasis "The elements must be arranged as required by the claim, but this is not an ipsissimis verbis test", i.e., word for word, Verbatim test!

The "Matlab" reference is used to anticipate the complete details of the claim elements and the arrangement (configurations) of their essence (real meaning), but only as much claim detail as the claim solicits to the full scope and breath of the claim language, in light of the specification.

A. Keeping this in ones' mind, the Appellants argue, halfway down on page -5-, that "the Matlab reference does not describe the inventions of claims 5 and 58 because it does not describe

using the capabilities of Matlab to perform the steps recited in those claims. Among other things, the Matlab reference does not describe an ordered list of operations [that] contains the operations which if performed in order on the earlier state of the three dimensional mesh model would result in a current state of the three dimensional mesh model. While it may be possible for a user to create an M-file in Matlab that contained operations which if performed in order on a earlier state of a three dimensional mesh model would result in a current state of the three dimensional mesh model, Applicants respectfully state that there is no description in the Matlab reference of any user having done so."

By the Appellants own analysis of Matlab to their own claimed invention, the Appellants conclude by admission (While it may be possible for a user to create an M-file in Matlab that contained operations which if performed in order on a earlier state of a three dimensional mesh model would result in a current state of the three dimensional mesh model, Applicants respectfully state that there is no description in the Matlab reference of any user having done so) that Matlab meets the claimed invention, but lacks the element of any user having done so.

While the Appellants argue that there is no description in the Matlab reference of any user having done so, independent claims 5 and 58 do not mention or allude to in any form in any of its elements to any "user" whatsoever!

Via the broad metes and bounds of the claim language of independent claims 5 and 58, there is no limitation as to a "user". Given the language of claim 58, "instructions executed by a processor", one might presume the instructions, processor combination to be artificial intelligence "AI", rather then a user.

However, as given by argument, it must have been the Appellants intent to comprise within said claims a user having done so, even so Matlab does teach this on page ii, in 2nd and 3rd paragraphs at 'The purpose of this Primer is to help you begin to use MATLAB. It is not intended to be a substitute for the User's Guide and Reference Guide for MATLAB. The Primer can best be used hands-on. You are encouraged to work at the computer as you read the Primer and freely experiment with examples. This Primer, along with the on-line help facility, usually suffice for students in a class requiring use of MATLAB.'

Herein, 'you', 'hands-on' and 'students' correspond to "user" and 'freely experiment with examples' corresponds to the user using the examples in the Matlab Primer during a session while using Matlab. Thus the Matlab Primer encourages and motivates users to work the examples on a computer that has Matlab installed on it.

Furthermore "freely experiment" is not Thomas Edison inventing the light bulb, nor undue experimentation, it is an encouragement to the user to try various combinations of the examples given in the Matlab Primer. (See MPEP 2106.V.2 The fact that experimentation is complex, however, will not make it undue if a person of skill in the art typically engages in such complex experimentation.) Throughout the Matlab Primer the user is suggested to try this or try that combination, for example, see page 4, 2nd from bottom line "Try them."; page 7, last line in section (8.) "Try it."; page 8, line 6 and 4th line from bottom, "Try it."; page 15, halfway down "Try it."; page 15 discusses 3-D mesh surface plots and suggests the user "To preview some of these capabilities, enter the command demo and select some of the graphics options."; page 18, Primer suggests the user to draw (plot) the 3-D mesh $z = e^{-x2-y2}$ over the range [-2, 2] x [-2, 2] (try it).

In as much as the Appellants argue that Matlab does not perform the steps recited in those claims, independent claim 5 has no steps.

Just to highlight a point the Appellants have already made, and that is equating an M-file to containing operations which if performed in order on a earlier state of a three dimensional mesh model would result in a current state of the three dimensional mesh model. The concept of an M-file is given in section 12, beginning on page 9 of the Matlab reference; and managing M-files in section 14, beginning on page 13. Three dimensional (3-D) mesh is discussed in the beginning of section 18 on page 15 and page 18 at 3-D mesh and surface plots.

The Appellants further rest their argument on the Examiner using the Appellants' claims as a guide to picking and choosing the functionality of Matlab that might be used to carry out the steps in the claims. But the existence of such functionality is not sufficient to render the claim as anticipated, the steps must have actually been carried out as recited (i.e., as in claim 5) or instructions which define the steps to be carried out must have been stored on a computer readable medium (i.e., as in claim 58). Neither of those conditions is shown to have been met by the Matlab reference. The mere fact that a user could hypothetically have carried out the steps of the invention using the programming functionality of the Matlab system does not anticipate the claims. See, e.g., Perricone v. Medicis Pharm. Corp., 432 F.3d 1368, 1378 (Fed. Cir. 2005); Cf. MPEP § 2112.02 ("Prior art device anticipates a claimed process if the device carries out the process during normal operation" (emphasis added)).

Recall, supra, that "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference."

Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Matlab does not hypothetically teach that a user could have carried out the steps of the invention, but rather expressly and/or inherently describes making use of how it stores, saves and reloads sessions, variables, M-files, restores a workspace to its former state, 3-D mesh surface plots, and the execution of M-files and demo files.

Matlab anticipates claims 5 and 58 by expressly or inherently description as detailed supra for the rejection of claims 5 and 58.

The user is motivated to explore the application and functionality of Matlab as disclosed (Matlab - page ii, 2nd paragraph) by "You are encouraged to work at the computer as you read the Primer and freely experiment with examples." After all, the reference is a Matlab Primer for the normal use of Matlab applications, not tricks and special techniques. Recall, Perricone v. Medicis Pharm. Corp., 432 F.3d 1368, 1378 (Fed. Cir. 2005), ("Prior art device anticipates a claimed process if the device carries out the process during normal operation").

Furthermore recall that the user is encouraged to preview some of the 3-D mesh plot capabilities by entering the command demo and selecting some of the graphics options (see page 15, section 18. Graphics).

Suppose the independent claim read "a method of retrieving a stored copy of a matrix A = [4 5 6; 1 2 3; 7 8 9]" would Matlab not serve as disclosing this claim because the claim was used as a guide to picking and choosing the functionality of Matlab that might be used to carry out the steps in the claim and the existence of such functionality is not sufficient to render the claim as anticipated, the steps must have actually been carried out as recited, even though Matlab

discloses on page 1, A = [1 2 3; 4 5 6; 7 8 9]" and saving and loading the 3-by-3 matrix assigned to variable A. Viola, the Appellant would have an instant patent. Rearrange the 3-by-3 matrix six more time and receive another six patents. But no, this is not the case, not when one marks anticipation where if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference is used.

Actually the Matlab Primer encourages one to pick and choose various functionality by "freely experiment with examples" (page ii, 2nd paragraph), "try it" (page 4, 2nd from bottom line "Try them."; page 7, last line in section (8.) "Try it."; page 8, line 6 and 4th line from bottom, "Try it."; page 15, halfway down "Try it."; page 15 discusses 3-D mesh surface plots and suggests the user "To preview some of these capabilities, enter the command demo and select some of the graphics options."; page 18, Primer suggests the user to draw (plot) the 3-D mesh $z = e^{-x2-y2}$ over the range [-2, 2] x [-2, 2] (try it).

Finding the demarcation between using a claim as a guide to picking and choosing the functionality of a reference that might be used to carry out the steps in said claim wherein the existence of such functionality is not sufficient to render the claim as anticipated, and using anticipation where if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference is only enlightened by the normal use and application of the reference. Recall Perricone v. Medicis Pharm. Corp., ("Prior art device anticipates a claimed process if the device carries out the process during normal operation").

In the case of the reference Matlab Primer, it is a primer, a small introductory pamphlet on the subject describing its normal operation. Thus Matlab expressly and/or inherently describes

each and every element as set forth in the claim in a single prior art reference and therefore anticipates claims 5 and 58. Particularly when the Matlab Primer encourages the user to freely experiment with examples, to preview some of the 3-D mesh capabilities and to draw a 3-D mesh $z = e^{-x2-y2}$ over the range [-2, 2] x [-2, 2] (try it).

Clams 6-10 and 59-63 depend from rejected independent claims 5 and 58 respectively, and therefore are also unpatentable over Matlab.

B. Appellants' allegations that Matlab does not anticipate claims 2-4, 55-57 and 114-117 are the same as for claims 5 and 58 supra. However, Matlab expressly and/or inherently describes each and every element as set forth in claims 115, 116 and 117 in its normal use of operation as detailed supra for rejected claims 115, 116 and 117.

In claims 115 – 117, concerning first, second and third states, MatLab anticipates these claims by describing M-files wherein MatLab executes a sequence of statements stored in diskfiles. Saving the 3-D wire mesh surface described on page 18 as M-files and executing the M-files establish a first state (one prior to executing the M-file) and a second state (the result after executing the M-file). At the bottom of page 9, since M-files can reference other M-files, including referencing itself recursively, a plurality of sequential states result thereby generating a third, fourth, fifth … states, hence meeting the limitations of storing first state, operations performed results in a second state, storing operations (M-file), reapplying at least some operations resulting in a third state (an M-file can reference other M-files, including referencing itself recursively) for claims 115 – 117.

While the Appellants argue that the reference Matlab does not describe claims 115, 116 or 117 as actually being performed or assembled (page -7- of Appeal Brief), but it is claim 117 that lacks the substance of actually being performed. For example, the system comprising:

a computer module for storing a copy ...;

a computer module for performing operations ...;

a computer module for storing a record ...;

a computer module for reapplying ... at least some of the operations; does not actually perform the operations, thus claim 117 merely comprise computer modules that are for something and not necessarily actually doing anything. This arrangement of elements might just as well be sitting there with the switch turned off! Although the elements are for something, it doesn't perform them!

For claim 117 to actually perform the operation, the comprising should be written as:

a computer module that stores a copy ...;

a computer module that performs operations ...;

a computer module that stores a record ...;

a computer module that reapplies ... at least some of the operations;

or

a computer module storing a copy ...;

a computer module performing operations ...;

a computer module storing a record ...;

a computer module reapplying ... at least some of the operations;

Whereas Matlab expressly and/or inherently describes each and every element as set forth in claims 115, 116 and 117 in its normal use of operation (see rejection of claims 115-117 supra) and actually performs this by virtue of whereby the user is encouraged to work at the computer as you read the Primer and freely experiment with examples, and therefore expressly and/or inherently has the user performing the operations as detailed supra for rejected claims 115-117.

While the Appellants argue that there is no description in the Matlab reference of any user having done so, independent claims 115-117 do not mention or allude to in any form in any of its elements to any "user" whatsoever!

Via the broad metes and bounds of the claim language of independent claims 115-117, there is no limitation as to a "user". Given the language of claim 116, "instructions executed by a processor", one might presume the instructions, processor combination to be artificial intelligence "AI", rather then a user.

However, as given by argument, it must have been the Appellants intent to comprise within said claims a user having done so, even so, Matlab does teach this on page ii, in 2nd and 3rd paragraphs at 'The purpose of this Primer is to help you begin to use MATLAB. It is not intended to be a substitute for the User's Guide and Reference Guide for MATLAB. The Primer can best be used hands-on. You are encouraged to work at the computer as you read the Primer and freely experiment with examples. This Primer, along with the on-line help facility, usually suffice for students in a class requiring use of MATLAB.'

Herein, 'you', 'hands-on' and 'students' correspond to "user" and 'freely experiment with examples' corresponds to the user using the examples in the Matlab Primer during a session

while using Matlab. Thus the Matlab Primer encourages and motivates users to work the examples on a computer that has Matlab installed on it.

Furthermore "freely experiment" is not Thomas Edison inventing the light bulb, nor undue experimentation, it is an encouragement to the user to try various combinations of the examples given in the Matlab Primer. (See MPEP 2106.V.2 The fact that experimentation is complex, however, will not make it undue if a person of skill in the art typically engages in such complex experimentation.) Throughout the Matlab Primer the user is suggested to try this or try that combination, for example, see page 4, 2nd from bottom line "Try them."; page 7, last line in section (8.) "Try it."; page 8, line 6 and 4th line from bottom, "Try it."; page 15, halfway down "Try it."; page 15 discusses 3-D mesh surface plots and suggests the user "To preview some of these capabilities, enter the command demo and select some of the graphics options."; page 18, Primer suggests the user to draw (plot) the 3-D mesh $z = e^{-x2-y2}$ over the range [-2, 2] x [-2, 2] (try it).

In as much as the Appellants argue that Matlab does not perform the steps recited in those claims, independent claims 115 and 117 have no steps.

Just to highlight a point the Appellants have already made with regard to claim 115, and that is the Matlab system as being capable of storing a list of operations (M-files) and as being capable of rendering mesh surface plots. The concept of an M-file is given in section 12, beginning on page 9 of the Matlab reference; and managing M-files in section 14, beginning on page 13. Three dimensional (3-D) mesh is discussed in the beginning of section 18 on page 15 and page 18 at 3-D mesh and surface plots. The full rejection to claims 115-117 is given supra.

Although the Appellants allege (bottom of page -7-) that the Matlab reference does not describe storing instructions on a computer readable medium to do so (i.e., as in claim 116), or assembling computer modules to do so (i.e., as in claim 117), Matlab counters this with attention focused to page 1 in section 2 at "Matrices can be introduced into MATLAB in several different ways: ... Created in a diskfile with your local editor"; page 2, 2nd paragraph at "If this file is named, say, data.ext (where .ext is any extension), the MATLAB command **load data.ext** will read this file to the variable data in your MATLAB workspace. This may also be done with script files (see section 12)."; page 9, section 12, bottom at "However, it is usually easier to use the MATLAB function load (see section 2)."page 13, last paragraph at "M-files must be in a directory accessible to MATLAB. On most mainframe or workstation network installations, personal M-files which are stored in a subdirectory of one's home directory named **matlab** will be accessible to MATLAB from any directory in which one is working."

Thus, diskfiles and M-files are stored instructions on computer readable medium and readable (accessible) by a computer (MATLAB session on a personal computer). Wherein diskfiles and M-files correspond to computer modules and are able to be assembled via the edit (!ed) command (see page 13, section 14. Managing M-files); and page 9, at '12. M-files.

MATLAB can execute a sequence of statements stored in diskfiles. Such files are called \M-files" because they must have the file type of \.m" as the last part of their filename. Much of your work with MATLAB will be in creating and refining M-files. M-files are usually created using your local editor.

The Appellants further rest their argument on the Examiner using the Appellants' claims as a guide to picking and choosing the functionality of Matlab that might be used to carry out the limitation of the claims. The mere fact that a user could hypothetically have carried out or assembled the invention using the programming functionality of the Matlab system does not anticipate the claims. See, e.g., Perricone v. Medicis Pharm. Corp., 432 F.3d 1368, 1378 (Fed. Cir. 2005); Cf. MPEP § 2112.02 ("Prior art device anticipates a claimed process if the device carries out the process during normal operation" (emphasis added)).

Recall, supra, that "A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." Verdegaal Bros. v. Union Oil Co. of California, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987).

Matlab does not hypothetically teach that a user could have carried out or assembled the invention, but rather expressly and/or inherently describes making use of how it stores, saves and reloads sessions, variables, M-files, restores a workspace to its former state, 3-D mesh surface plots, and the execution of M-files and demo files.

Matlab anticipates claims 115-117 by expressly and/or inherently description as detailed supra for the rejection of claims 115-117.

The user is motivated to explore the application and functionality of Matlab as disclosed (Matlab - page ii, 2nd paragraph) by "You are encouraged to work at the computer as you read the Primer and freely experiment with examples." After all, the reference is a Matlab Primer for the normal use of Matlab applications, not tricks and special techniques. Recall, Perricone v.

Medicis Pharm. Corp., 432 F.3d 1368, 1378 (Fed. Cir. 2005), ("Prior art device anticipates a claimed process if the device carries out the process during normal operation").

Again suppose the independent claim read "a method of retrieving a stored copy of a matrix A = [2 3;4 5] + [i* [6 7;8 9]" would Matlab not serve as disclosing this claim because the claim was used as a guide to picking and choosing the functionality of Matlab that might be used to carry out the steps in the claim and the existence of such functionality is not sufficient to render the claim as anticipated, the steps must have actually been carried out as recited, even though Matlab discloses on page 1, A = [1 2;3 4] + [i* [5 6;7 8]" and saving and loading the 3-by-3 matrix assigned to variable A. Viola, the Appellant would have an instant patent. Rearrange the 3-by-3 matrix seven more time and receive another seven patents. But no, this is not the case, not when one marks anticipation where if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference is used.

Actually the Matlab Primer encourages one to pick and choose various functionality by "freely experiment with examples" (page ii, 2nd paragraph), "try it" (page 4, 2nd from bottom line "Try them."; page 7, last line in section (8.) "Try it."; page 8, line 6 and 4th line from bottom, "Try it."; page 15, halfway down "Try it."; page 15 discusses 3-D mesh surface plots and suggests the user "To preview some of these capabilities, enter the command demo and select some of the graphics options."; page 18, Primer suggests the user to draw (plot) the 3-D mesh $z = e^{-x2-y2}$ over the range [-2, 2] x [-2, 2] (try it).

Claims 2-4, 55-57 and 114 depend from rejected claims 115-117 respectively, and therefore are also unpatentable over Matlab.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Gregory F. Cunningham, Examiner G.A.U. 2624

J. F. Cunningham 1/16/08

Conferees:

Matthew C. Bella, SPE G.A.U. 2624

MATTHEW C. BELLA
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2600

Eileen D. Lillis, SPE G.A.U. 2624

Gregory F. Cunningham, Examiner G.A.U. 2624